PRACTICE WORKBOOK

MACHINE LEARNING

KISHAU ROGERS

- Background: Computer Science, Entrepreneur, 24yrs delivering software solutions
- Blog: www.bigthinking.io
- Email: kishau@bigthinking.io
- TwitterS: @kishau, @bigthinkingio
- Current Focus: Machine Learning @ time:study

Weill Cornell Medicine
NewYork-Presbyterian
Stony Brook Medicine
University of Colorado Hospital
UNIVERSITY OF COLORADO HEALTH
COLUMBIA UNIVERSITY MEDICAL CENTER
SOUTHAMPTON HOSPITAL
MedStar Washington Hospital Center

TO PRACTICE: USE AZURE ML STUDIO

Using Azure ML Studio

1. In your Web browser, navigate to http://studio.azureml.net and click the **Sign Up** button.
2. Click **Sign In under Free Workspace**. Then sign in using your Microsoft account.

Walkthrough Step 1: Create a Machine Learning workspace

03/23/2017 • 2 minutes to read • Contributors

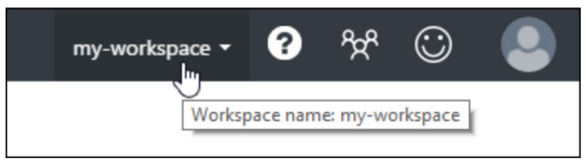
This is the first step of the walkthrough, Develop a predictive analytics solution in Azure Machine Learning.

1. **Create a Machine Learning workspace**
2. Upload existing data
3. Create a new experiment
4. Train and evaluate the models
5. Deploy the Web service
6. Access the Web service

To use Machine Learning Studio, you need to have a Microsoft Azure Machine Learning workspace. This workspace contains the tools you need to create, manage, and publish experiments.

The administrator for your Azure subscription needs to create the workspace and then add you as an owner or contributor. For details, see Create and share an Azure Machine Learning workspace.

After your workspace is created, open Machine Learning Studio (https://studio.azureml.net/Home). If you have more than one workspace, you can select the workspace in the toolbar in the upper-right corner of the window.



QUIZ: IDENTIFY A ML PROJECT

ML?	PROJECT DESCRIPTION	DATA
	Classify store products into one of 10 product categories.	Inventory/Product data sheet
	Predict which blog articles will be popular.	Historical Article & Reader Statistics
	Compute the mean wage of your employees.	Payroll & Salary Info
	Given various measurements of wine, predict the origin of the wine.	Wine Chemistry Data
	Detect if customer support engagement is hostile	Historical Support Logs with Hostility Labels
	Classify email messages as spam or non-spam	Historical Email Logs with Spam Flags
	Recommending the two best designers based on job requirements	Resumes & Job Listing

USE CASE

CUSTOMER RETENTION

EXERCISE 1: PROJECT INITIATION BLUEPRINT

COMPLETE PROJECT INITIATION EXERCISE

http://bigthinking.io/btFiles/training/MLIntroWorkBook.pdf

TASK:

- ☐ Review the “Data DataSheet” (next page)
- ☐ Identify a well-formed question that can be answered with the dataset(s) provided.
- ☐ Select the appropriate machine learning type

QUESTION TO BE ANSWERED			
LEARNING TYPE	SUPERVISED	UNSUPERVISED	REINFORCED

EXERCISE 1: DATA DATASHEET (LOAN DATA)

Abstract: Using historical customer data, predict customer retention (or churn)

Filename: http://bigthinking.io/btFiles/training/data/CustomerChurnClean.csv

Dataset Characteristics	Multivariate	**Instances**	7044	**Sector**	Telecommunications
Attribute Characteristics	Integer,Char	**Attributes**	21	**Source Date**	Sample

Dataset Information: Customer data for a Telecommunications company.

- customerID - Customer Identifier
- genderCustomer - Customer gender (female, male)
- SeniorCitizen - Is the customer a senior citizen (1, 0)
- Partner - Whether the customer has a partner or not (Yes, No)
- Dependents - Whether the customer has dependents or not (Yes, No)
- tenure - Number of months the customer has stayed with the company
- PhoneService - Whether the customer has a phone service or not (Yes, No)
- MultipleLines - Whether the customer has multiple lines or not (Yes, No, No phone service)
- InternetService - Customer's internet service provider (DSL, Fiber optic, No)
- OnlineSecurity - Whether the customer has online security or not (Yes, No, No internet service)
- OnlineBackup - Whether the customer has online backup or not (Yes, No, No internet service)
- DeviceProtection - Whether the customer has device protection or not (Yes, No, No internet service)
- TechSupport - Whether the customer has tech support or not (Yes, No, No internet service)
- StreamingTV - Whether the customer has streaming TV or not (Yes, No, No internet service)
- StreamingMovies - Whether the customer has streaming movies or not (Yes, No, No internet service)
- Contract - The contract term of the customer (Month-to-month, One year, Two year)
- PaperlessBilling - Whether the customer has paperless billing or not (Yes, No)
- PaymentMethod - The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- MonthlyCharges - The amount charged to the customer monthly
- TotalCharges - The total amount charged to the customer
- Churn - Whether the customer churned or not (Yes or No)

EXERCISE 2: DATA ACQUISITION

Task: Import your training dataset

1. Sign-in ML Studio (http://studio.azureml.net)

2. Create a NEW blank Experiment (https://docs.microsoft.com/en-us/azure/machine-learning/studio/walkthrough-3-create-new-experiment)

3. To access an online data source to your ML Studio experiment, find the Import Data module by searching for “Import” on the left-hand menu

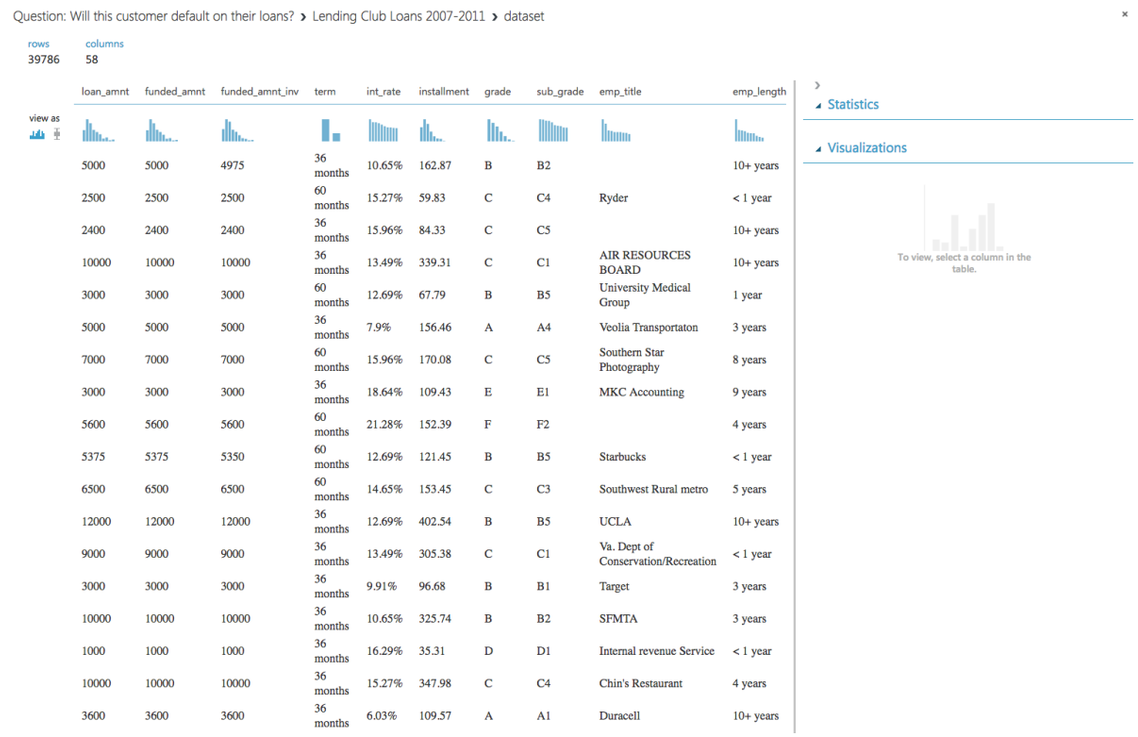
4. Drag the Import Data model onto your canvas

5. Then provide the following parameters needed to access the data.

   - Source: Web URL via HTTP
   - Format: CSV (with header)
   - URL: http://bigthinking.io/btFiles/training/data/CustomerChurnClean.csv (pre-processed) Original can be found at /CustomerChurn.csv.

6. Click RUN

7. To confirm your data import, right quick on the Import Data port and select “Visualize”



EXERCISE 3: FEATURE SELECTION

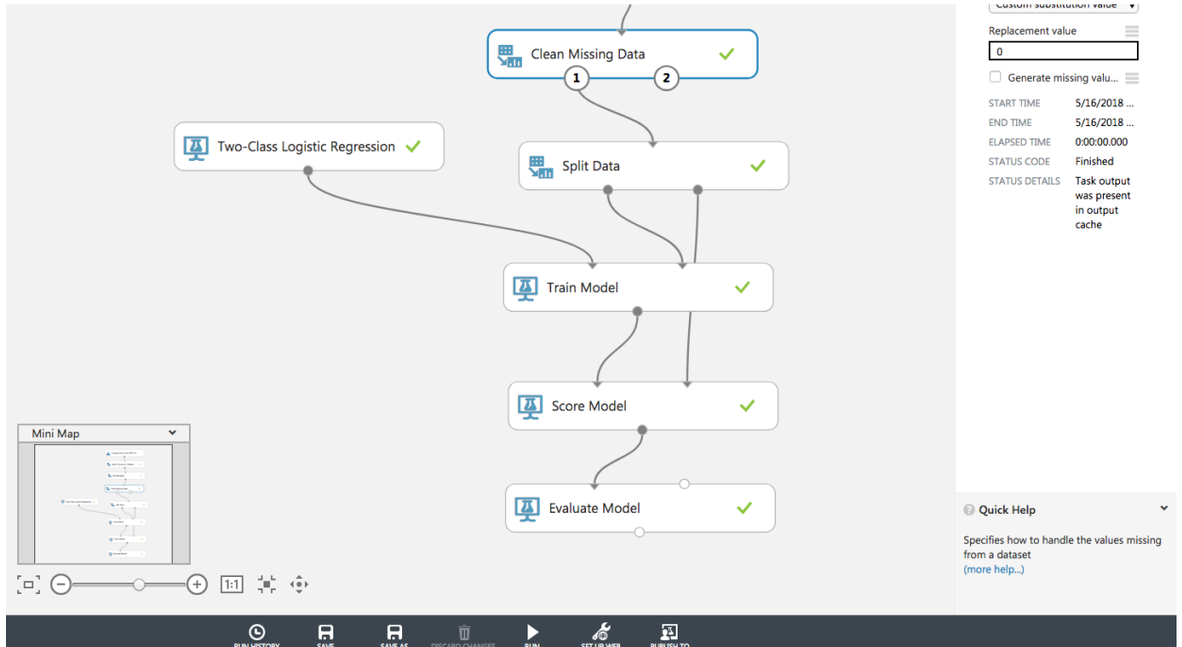
Task: Select the data to include in your experiment

1. Add the Select Columns in Dataset module to your experiment, and connect to your imported dataset

2. Use the Launch Column Selector to Identify the Columns to be included (or excluded),

3. Select RUN

4. To confirm your selection, right quick on the Select Columns in Dataset output port and select “Visualize”



EXERCISE 4: DATA CLEANING

Task: Clean your training dataset

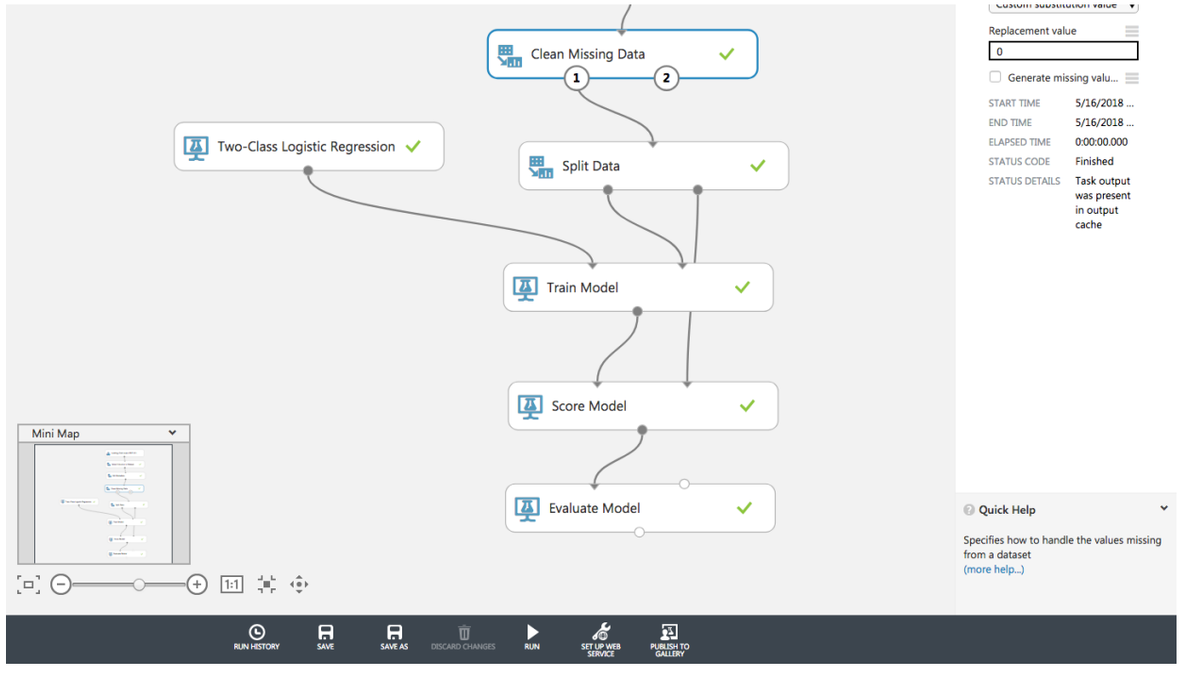
1. Add the Clean Missing Data module to your experiment, and connect to your dataset (the output port for “Select Columns in Dataset”)

2. Determine the Columns to be cleaned, choose the columns that contain the missing values you want to change.

3. Determine the cleaning method (remove, replace etc).

4. Update the Clean Missing Data module settings per your answers to #2 and #3

5. Select RUN



EXERCISE 5: MODEL BUILDING

Task: Classify customers into two groups (“will pay fully” vs “will default”)

1. Split your dataset into a Training/Test

2. Add the Two Class Logistic Regression module to the experiment.

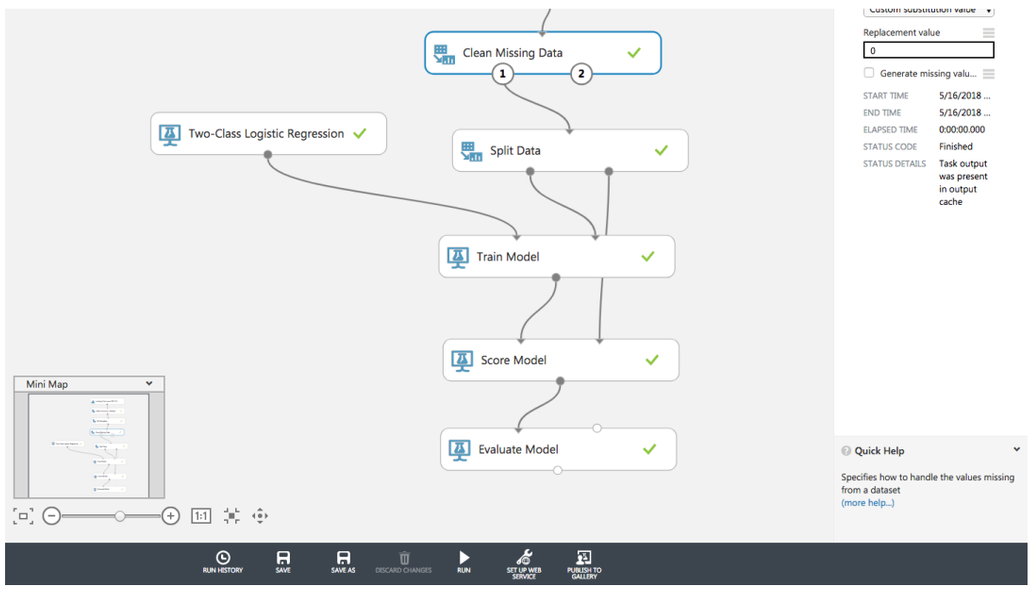
3. Specify how you want the model to be trained, by setting the Create trainer mode option (Single Parameter)

4. Add the Train Model module.

5. On the left input, attach the untrained mode. Attach the training dataset to the right-hand input of Train Model.

6. Choose the column that contains outcomes the model can use for training (Label)

7. Run the experiment.



EXERCISE 6: MODEL EVALUATION

Task: Evaluate model performance

1. Add the Score Model module to your experiment.

2. Attach the trained model and a dataset containing new input data (split data).

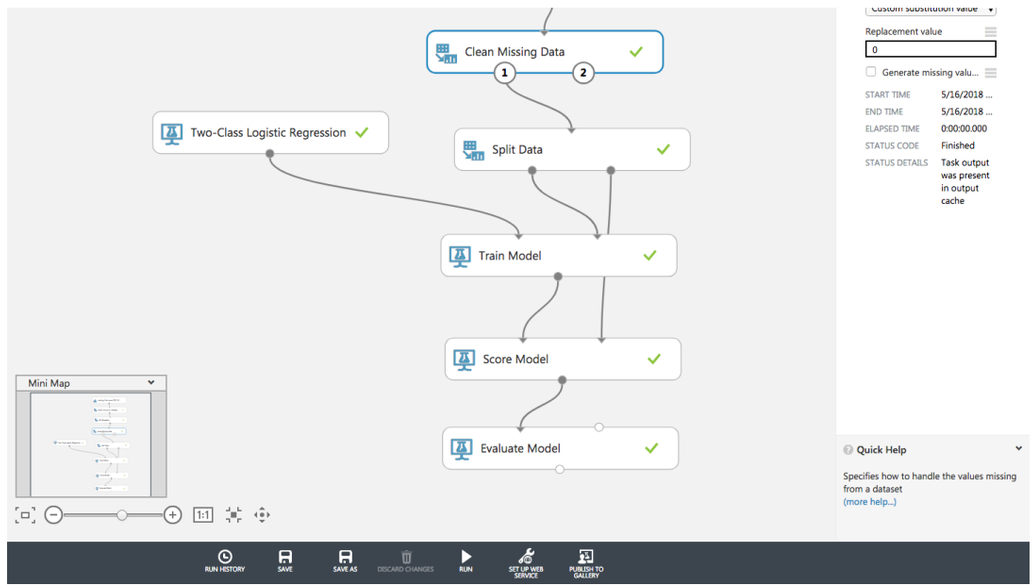
3. Add the Evaluate Model module to your experiment.

4. Connect the Scored dataset output of the Score Model to the input of Evaluate Model.

5. Connect the output of the Split Data module that contains the testing data to the right-hand input of Evaluate Model.

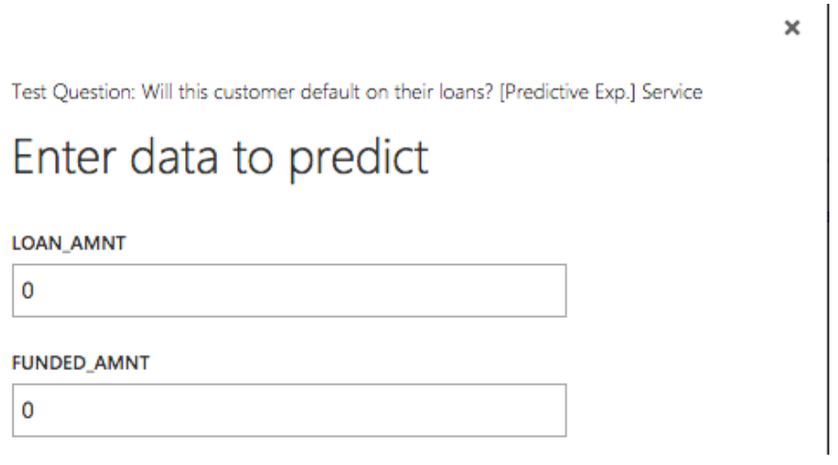
6. Run the experiment.

7. After you run Evaluate Model, right-click the module and select Evaluation results to see the results



EXERCISE 7: DEPLOY

Task: Deploy a web service that receives new input data and returns the predicted creditworthiness (will the customer repay the loan?)



1. To convert your training experiment to a predictive experiment by clicking **Run** at the bottom of the experiment canvas, click **Set Up Web Service**, then select **Predictive Web Service**.

2. To deploy your predictive experiment, click **Run** at the bottom of the experiment canvas. Once the experiment has finished running, click **Deploy Web Service** and select **Deploy Web Service Classic.**

3. To test the Request Response web service, click the Test button in the web service dashboard. A dialog pops up to ask you for the input data for the service. These are the columns expected by the scoring experiment. Enter a set of data and then click OK. The results generated by the web service are displayed at the bottom of the dashboard.

BONUS

4. Add a script (R) for returning ONLY the predicted customer status (retained/churn) label (as opposed to the full results returned by the Scored Model)

USE CASE

CREDITWORTHINESS

EXERCISE 1: PROJECT INITIATION BLUEPRINT

COMPLETE PROJECT INITIATION EXERCISE

http://bigthinking.io/btFiles/training/MLIntroWorkBook.pdf

TASK:

- ☐ Review the “Data DataSheet” (next page)
- ☐ Identify a well-formed question that can be answered with the dataset(s) provided.
- ☐ Select the appropriate machine learning type

QUESTION TO BE ANSWERED			
LEARNING TYPE	SUPERVISED	UNSUPERVISED	REINFORCED

EXERCISE 1: DATA DATASHEET (LOAN DATA)

Abstract: Using historical loan data, classify loan applicants by creditworthiness

Filename: http://bigthinking.io/btFiles/training/data/LoanData.csv

Dataset Characteristics	Multivariate	Instances	39786	Area	
Attribute Characteristics	Integer,Char	Attributes	58	Source Date	2007-2011

Dataset Information: This dataset includes statistics on all loans issued by Lending Club during the period 2007-2011.

- loan_amnt - Loan amount
- term - Payment Term
- int_rate - Interest Rate
- emp_length - Employment length
- zip_code - first 3 digits of the applicant's zipcode
- addr_state - State of address
- home_ownership - Home ownership status (OWN, RENT, MORTAGE)
- annual_inc - Annual Income
- loan_status - Loan Status (Fully Paid, Charged Off)
- delinq_2yrs - Delinquencies in the past 2 years (prior to application)
- inq_last_6mnths - Inquiries in the last 6 months (prior to application)
- mths_since_last_delinq - Number of months since the last delinquency.
- open_acc - Number of open credit accounts
- revol_util - % of revolving credit used

EXERCISE 2: DATA ACQUISITION

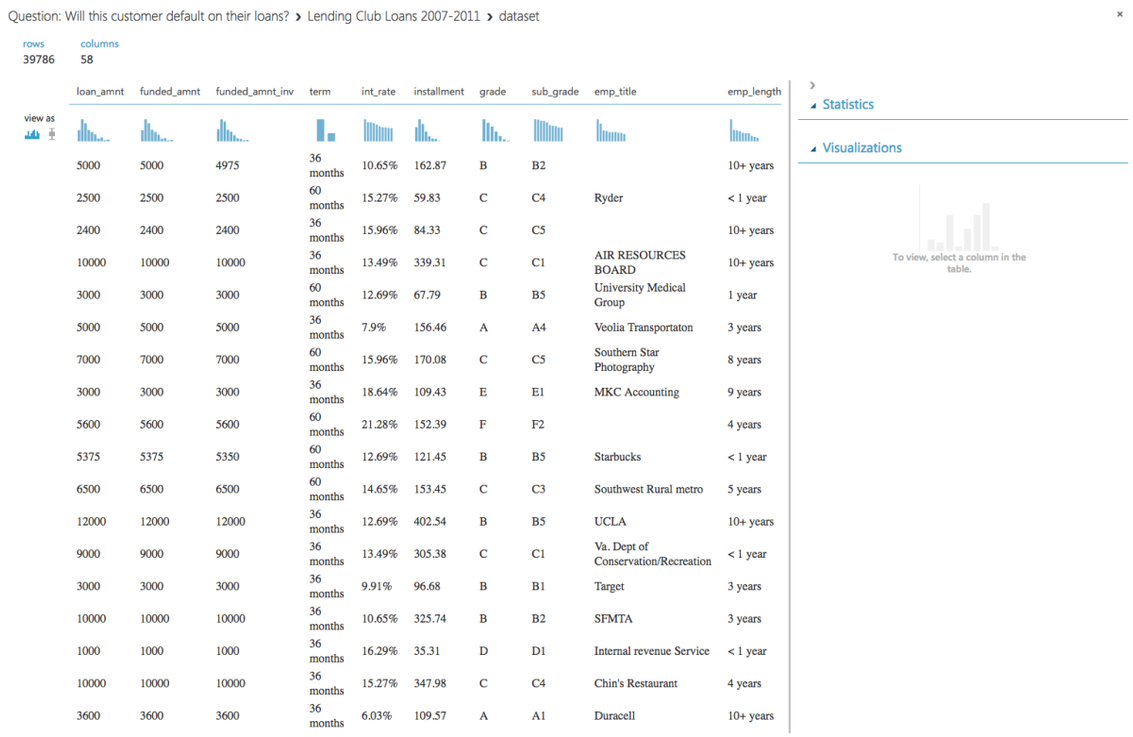
Task: Import your training dataset

1. Sign-in ML Studio (http://studio.azureml.net)

2. Create a NEW blank Experiment (https://docs.microsoft.com/en-us/azure/machine-learning/studio/walkthrough-3-create-new-experiment)

3. To access an online data source to your ML Studio experiment, find the Import Data module by searching for “Import” on the left-hand menu

4. Drag the Import Data model onto your canvas



5. Then provide the following parameters needed to access the data.

   - Source: Web URL via HTTP
   - Format: CSV (with header)
   - URL: http://bigthinking.io/btFiles/training/data/LoanData.csv

6. Click RUN

7. To confirm your data import, right quick on the Import Data port and select “Visualize”

EXERCISE 3: FEATURE SELECTION

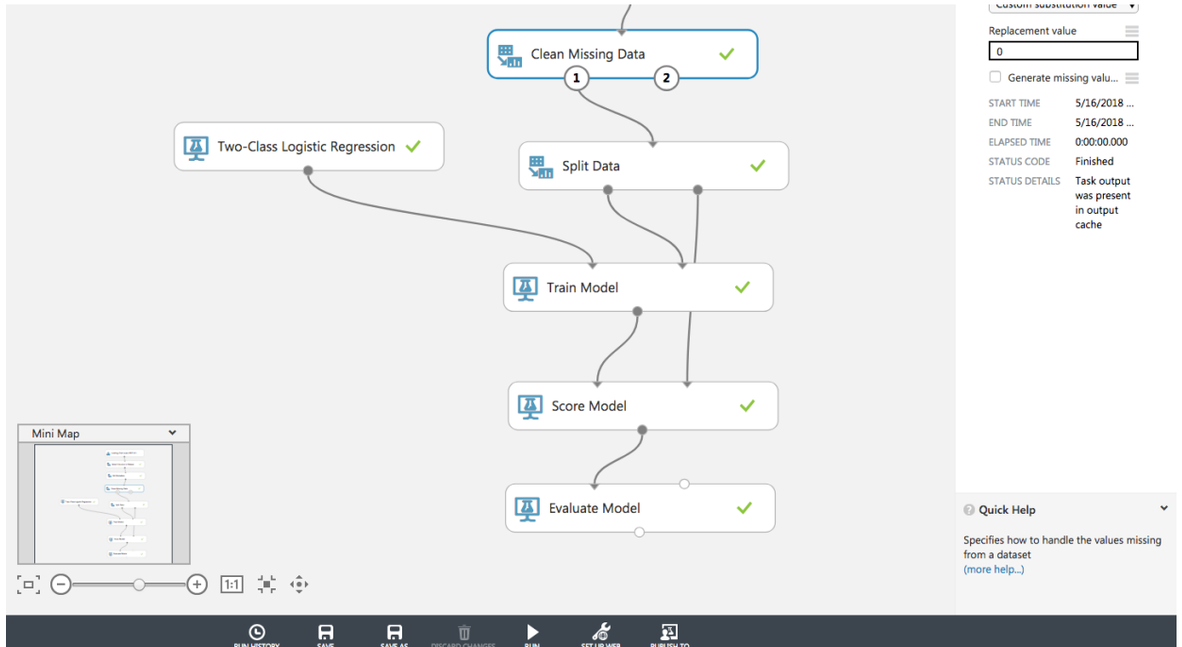
Task: Select the data to include in your experiment

1. Add the Select Columns in Dataset module to your experiment, and connect to your imported dataset

2. Use the Launch Column Selector to Identify the Columns to be included (or excluded),

3. Select RUN

4. To confirm your selection, right quick on the Select Columns in Dataset output port and select “Visualize”



EXERCISE 4: DATA CLEANING

Task: Clean your training dataset

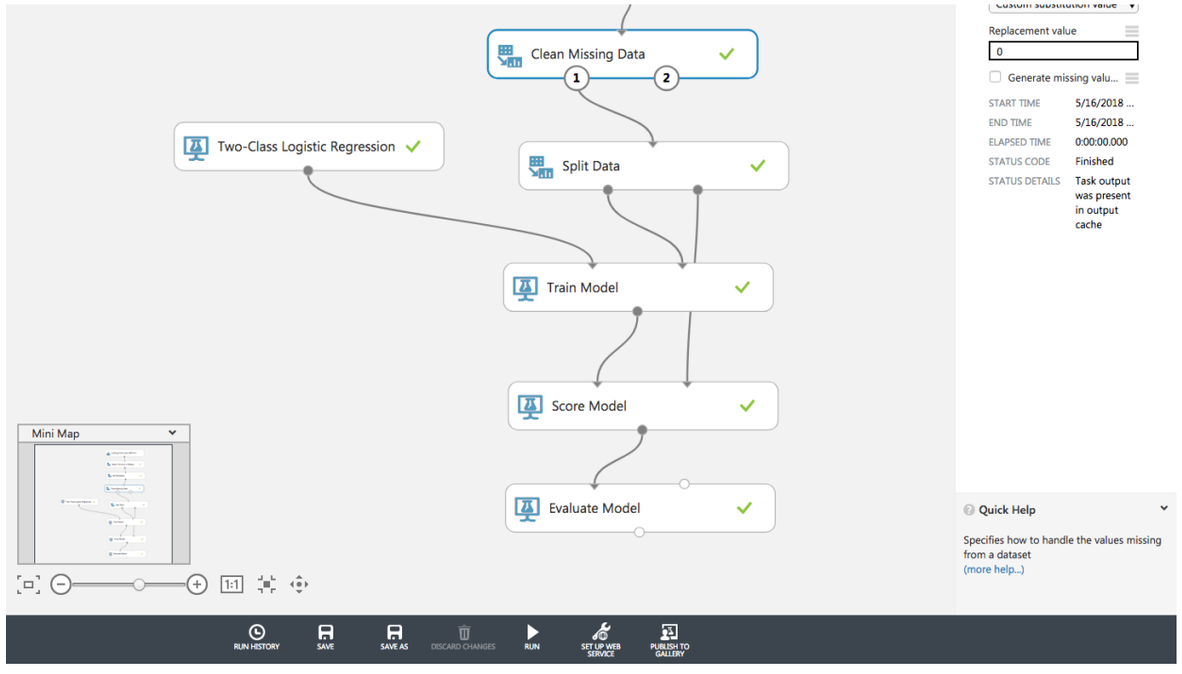
1. Add the Clean Missing Data module to your experiment, and connect to your dataset (the output port for “Select Columns in Dataset”)

2. Determine the Columns to be cleaned, choose the columns that contain the missing values you want to change.

3. Determine the cleaning method (remove, replace etc).

4. Update the Clean Missing Data module settings per your answers to #2 and #3

5. Select RUN



EXERCISE 5: MODEL BUILDING

Task: Classify customers into two groups (“will pay fully” vs “will default”)

1. Split your dataset into a Training/Test

2. Add the Two Class Logistic Regression module to the experiment.

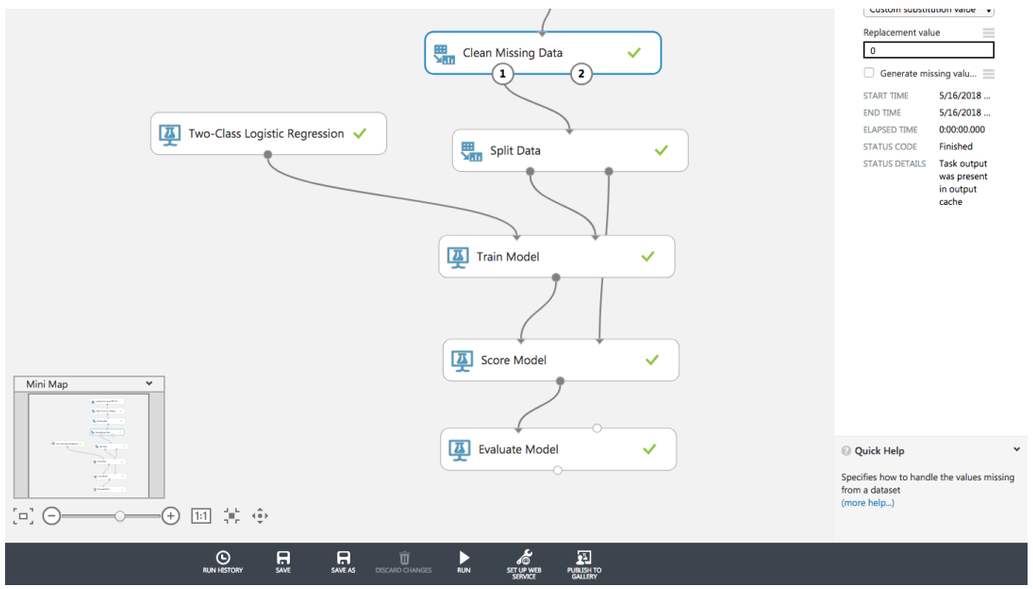
3. Specify how you want the model to be trained, by setting the Create trainer mode option (Single Parameter)

4. Add the Train Model module.

5. On the left input, attach the untrained mode. Attach the training dataset to the right-hand input of Train Model.

6. Choose the column that contains outcomes the model can use for training (Label)

7. Run the experiment.



EXERCISE 6: MODEL EVALUATION

Task: Evaluate model performance

1. Add the Score Model module to your experiment.

2. Attach the trained model and a dataset containing new input data (split data).

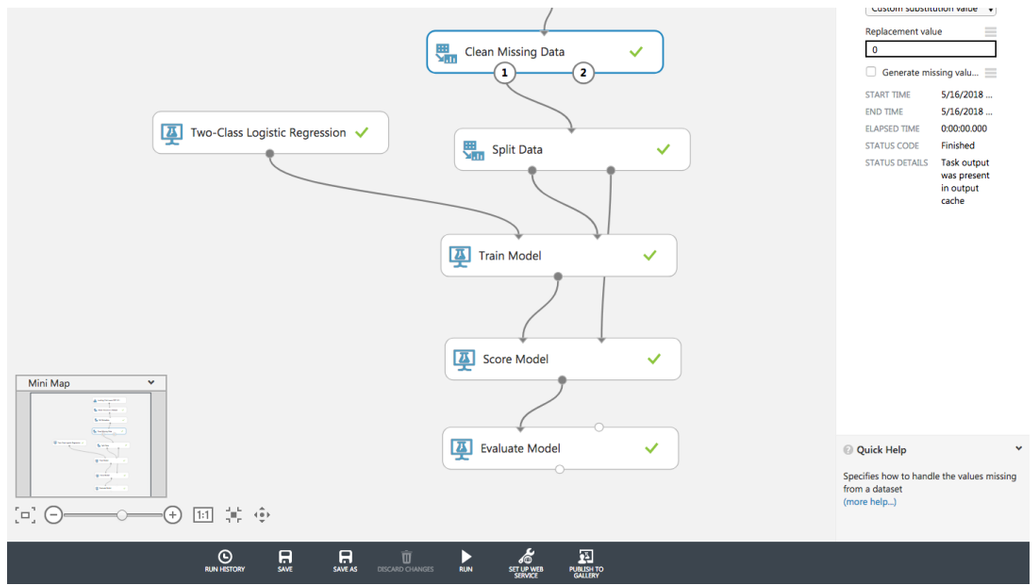
3. Add the Evaluate Model module to your experiment.

4. Connect the Scored dataset output of the Score Model to the input of Evaluate Model.

5. Connect the output of the Split Data module that contains the testing data to the right-hand input of Evaluate Model.

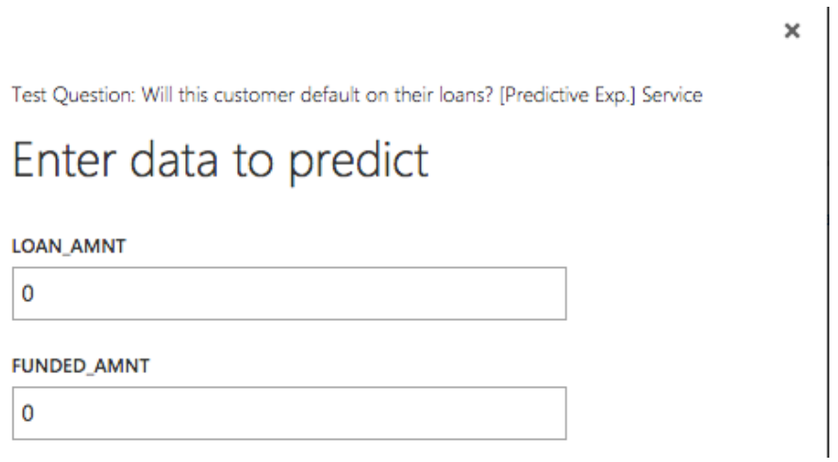
6. Run the experiment.

7. After you run Evaluate Model, right-click the module and select Evaluation results to see the results



EXERCISE 7: DEPLOY

Task: Deploy a web service that receives new input data and returns the predicted creditworthiness (will the customer repay the loan?)



1. To convert your training experiment to a predictive experiment by clicking **Run** at the bottom of the experiment canvas, click **Set Up Web Service**, then select **Predictive Web Service**.

2. Add a script (R) for returning ONLY the predicted creditworthiness label (as opposed to the full results returned by the Scored Model)

3. To deploy your predictive experiment, click **Run** at the bottom of the experiment canvas. Once the experiment has finished running, click **Deploy Web Service** and select **Deploy Web Service Classic.**

4. To test the Request Response web service, click the Test button in the web service dashboard. A dialog pops up to ask you for the input data for the service. These are the columns expected by the scoring experiment. Enter a set of data and then click OK. The results generated by the web service are displayed at the bottom of the dashboard.

USE CASE

EMPLOYEE TIME

EXERCISE : PROJECT INITIATION BLUEPRINT

COMPLETE PROJECT INITIATION EXERCISE

http://bigthinking.io/btFiles/training/MLIntroWookBook.pdf

TASK:

- ☐ Review the “Data DataSheet” (next page)
- ☐ Identify a well-formed question that can be answered with the dataset(s) provided.
- ☐ Select the appropriate machine learning type

QUESTION TO BE ANSWERED			
LEARNING TYPE	SUPERVISED	UNSUPERVISED	REINFORCED

DATA DATASHEET 1: TIME & EFFORT DATA

Abstract: Using historical time & effort data, classify employees by role

Filename: http://bigthinking.io/btFiles/training/data/PersonWorkDayStyleSampleND.csv

Dataset Characteristics	Multivariate	**Instances**	1500	**Area**	
Attribute Characteristics	Integer	**Attributes**	6	**Source Date**	2016

Dataset Information: These data represent a random sample of employee time reported during 2016. The data provides the total daily hours reported across four (4) common activities. Each instance represents a person's "work day". Preprocessing: aggregating related work activities into four major categories.

- MaskedPersonID (int): Unique person ID
- Administrative (int): Daily Administrative Hours
- Teaching (int): Daily hours spent Teaching
- Service (int): Delivery: Daily Hours spent on Professional Service Delivery
- Other (int): Daily Hours spent doing other things
- WorkStyle (char): Classification to be used for training (determined by role/title)

EXERCISE: DATA ACQUISITION

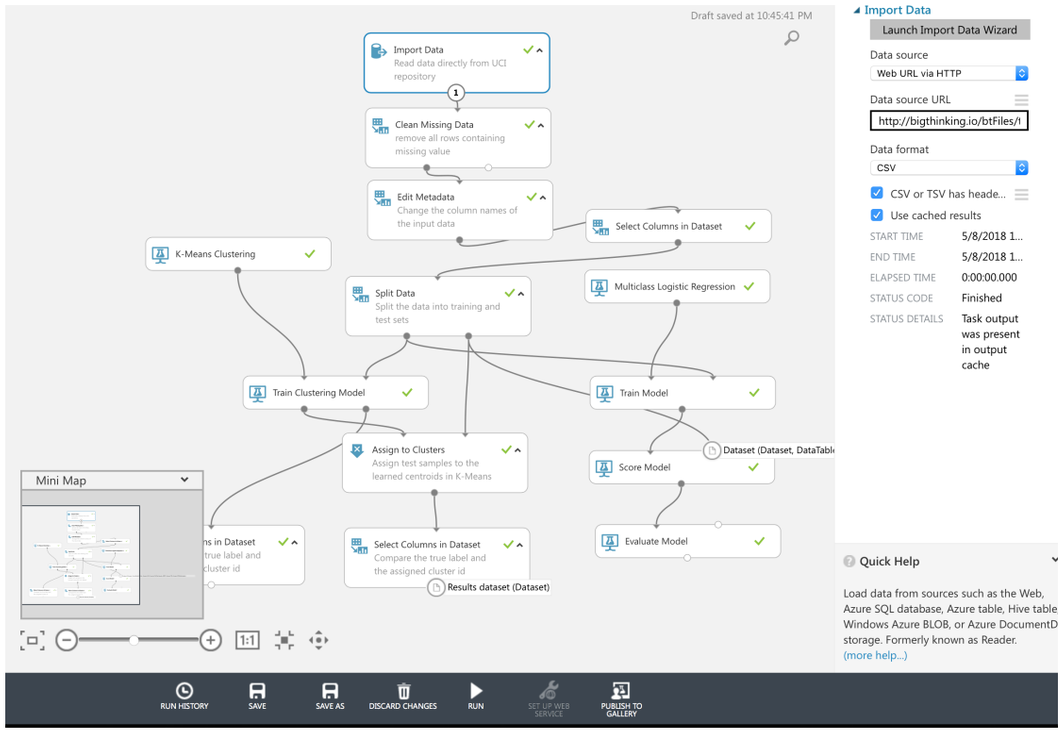
Task: Import your training dataset

1. Sign-in ML Studio (http://studio.azureml.net)

2. Create a NEW blank Experiment (https://docs.microsoft.com/en-us/azure/machine-learning/studio/walkthrough-3-create-new-experiment)

3. To access an online data source to your ML Studio experiment, find the Import Data module by searching for “Import” on the left-hand menu

4. Drag the Import Data model onto your canvas

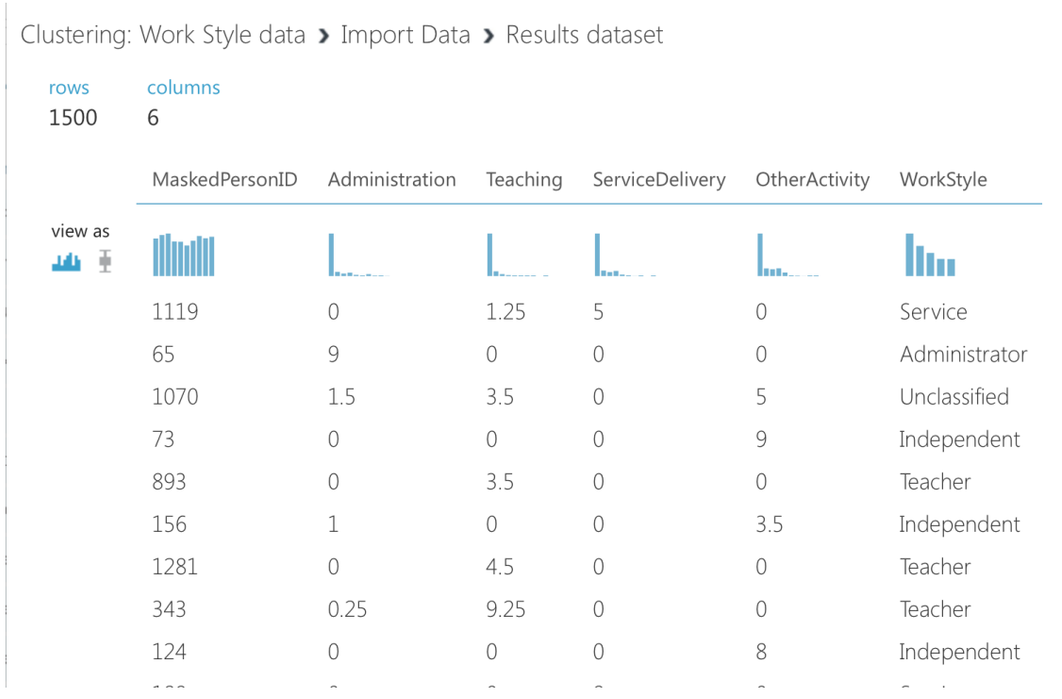


5. Then provide the following parameters needed to access the data.

   - Source: Web URL via HTTP
   - Format: CSV (with header)
   - URL: http://bigthinking.io/btFiles/training/data/PersonWorkDayStyleSampleND.csv

6. Click RUN

7. To confirm your data import, right quick on the Import Data port and select “Visualize”



MaskedPersonID	Administration	Teaching	ServiceDelivery	OtherActivity	WorkStyle
1119	0	1.25	5	0	Service
65	9	0	0	0	Administrator
1070	1.5	3.5	0	5	Unclassified
73	0	0	0	9	Independent
893	0	3.5	0	0	Teacher
156	1	0	0	3.5	Independent
1281	0	4.5	0	0	Teacher
343	0.25	9.25	0	0	Teacher
124	0	0	0	8	Independent

EXERCISE: DATA CLEANING

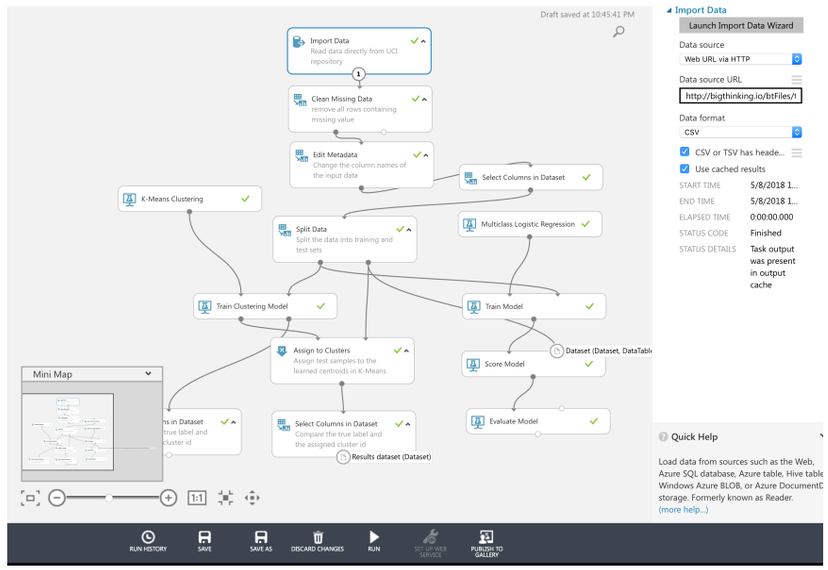
Task: Clean your training dataset

1. Add the Clean Missing Data module to your experiment, and connect to your imported dataset

2. Determine the Columns to be cleaned, choose the columns that contain the missing values you want to change.

3. Determine the cleaning method (remove, replace etc).

4. Update the Clean Missing Data module settings per your answers to #2 and #3



5. Select RUN

EXERCISE: MODEL BUILDING

Task: Classify work days by “work styles”

1. Split your dataset into a Training/Test

2. Add the Multiclass Logistic Regression module to the experiment.

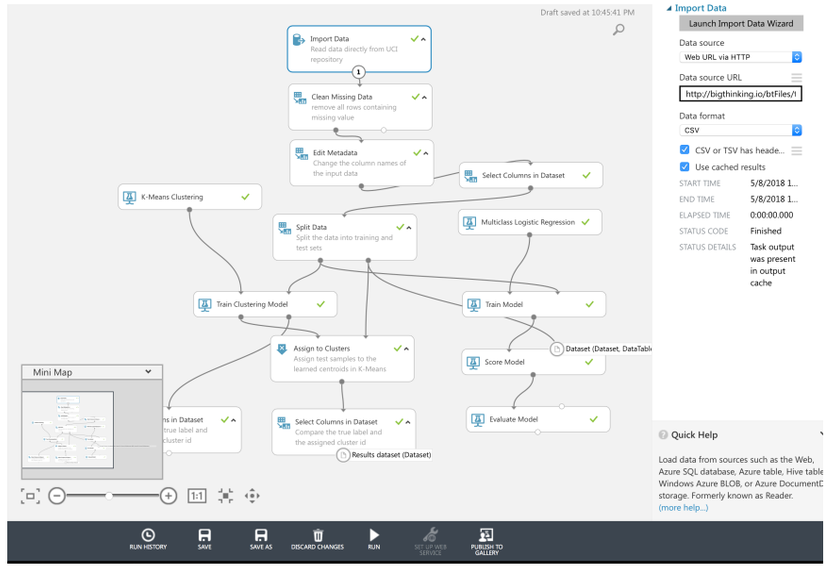
3. Specify how you want the model to be trained, by setting the Create trainer mode option (Single Parameter)

4. Add the Train Model module.

5. On the left input, attach the untrained mode. Attach the training dataset to the right-hand input of Train Model.

6. Choose the column that contains outcomes the model can use for training (Label)

7. Run the experiment.



EXERCISE: MODEL EVALUATION

Task: Evaluate model performance

1. Add the Score Model module to your experiment.

2. Attach the trained model and a dataset containing new input data (split data).

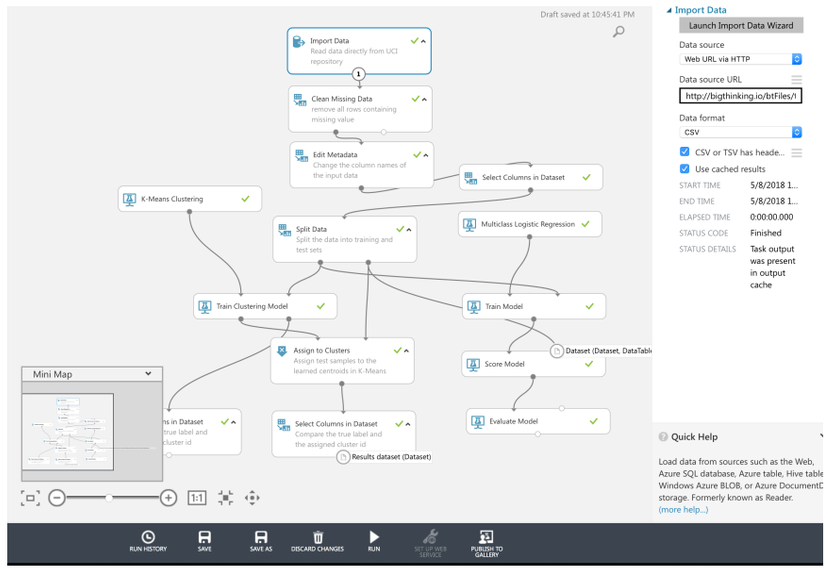
3. Add the Evaluate Model module to your experiment.

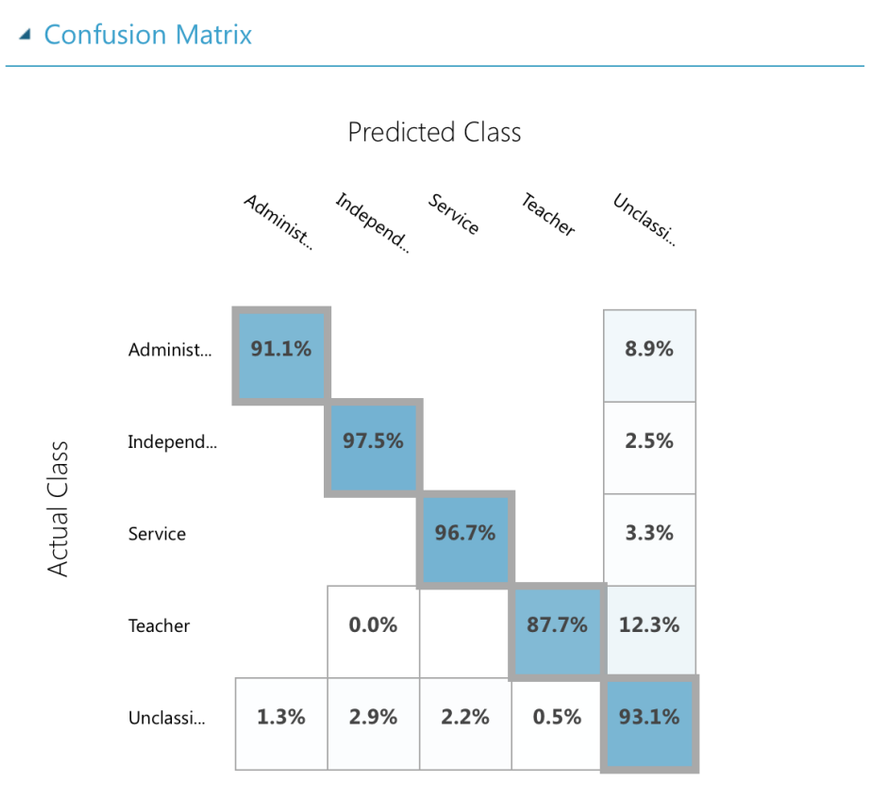
4. Connect the Scored dataset output of the Score Model to the input of Evaluate Model.

5. Connect the output of the Split Data module that contains the testing data to the right-hand input of Evaluate Model.

6. Run the experiment.

7. After you run Evaluate Model, right-click the module and select Evaluation results to see the results (confusion matrix).





EXERCISE: DEPLOY

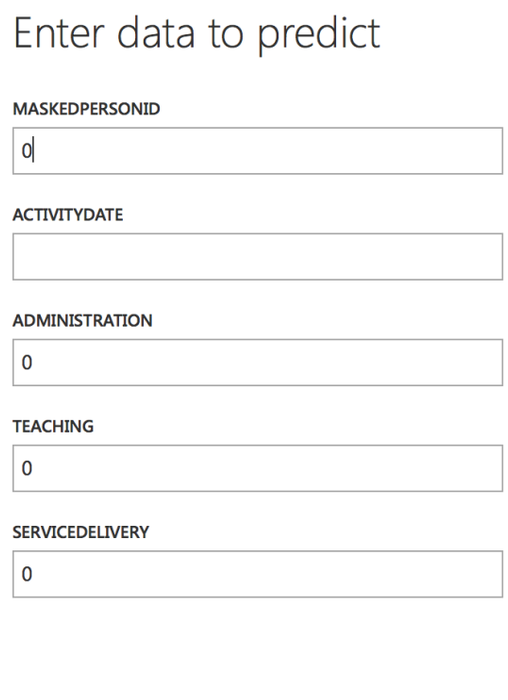
Task: Deploy a web service that receives new input data and returns the predicted "work style" and its probability score.

1. To convert your training experiment to a predictive experiment by clicking **Run** at the bottom of the experiment canvas, click **Set Up Web Service**, then select **Predictive Web Service**.

2. Add a script (R) for returning ONLY the predicted work style and its probability (as opposed to the full results returned by the Scored Model which includes the scores of all potential "work style" classifications)

3. To deploy your predictive experiment, click **Run** at the bottom of the experiment canvas. Once the experiment has finished running, click **Deploy Web Service** and select **Deploy Web Service Classic.**

4. To test the Request Response web service, click the Test button in the web service dashboard. A dialog pops up to ask you for the input data for the service. These are the columns expected by the scoring experiment. Enter a set of data and then click OK. The results generated by the web service are displayed at the bottom of the dashboard.



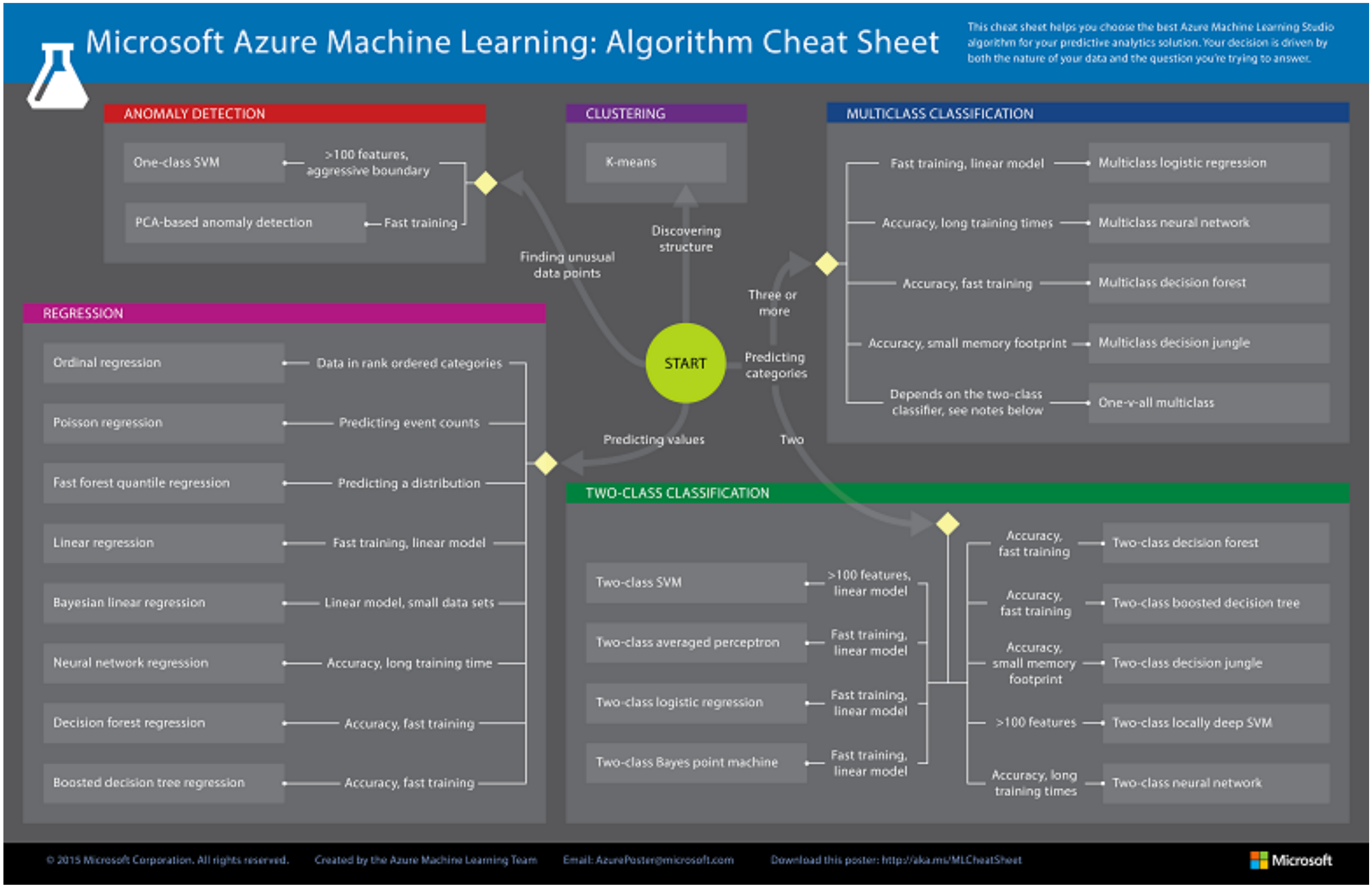

```
#Map optional input ports to variables
dataset <- maml.mapInputPort(1) # class:
data.frame

#Get the Scored Labels & their corresponding
probabilities
data.set = data.frame('Scored
Labels'=dataset['Scored Labels'], 'Label
Probability'=apply(dataset[,8:12],1,max))

#Select data.frame to be sent to the output
Dataset port
maml.mapOutputPort("data.set");
```

RESOURCES & WORKSHEETS

MORE MACHINE LEARNING RESOURCES

http://MachineLearning.bigthinking.io

MACHINE LEARNING : ALGORITHM GUIDE (FOR ML STUDIO)



Source: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet

MACHINE LEARNING : PROGRAMMING GUIDES

- R Cheat Sheets - https://www.rstudio.com/resources/cheatsheets/
- R Reference Card - https://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf
- Beginner's Python Cheat Sheet - http://bigthinking.io/btFiles/training/resources/beginners_python_cheat_sheet_pcc_all.pdf

MACHINE LEARNING : PROJECT INITIATION BLUEPRINT

QUESTION TO BE ANSWERED			
LEARNING TYPE	SUPERVISED	UNSUPERVISED	REINFORCED
DATA		**MODEL**	
QA		**DEPLOY**	
MONITOR			

SYSTEMS THINKING FRAMEWORK FOR MACHINE LEARNING

ML PROCESS	DISCOVER	ACQUIRE	PREPARE	BUILD	VALIDATE	DEPLOY	MONITOR
GOAL	Identify hypothesis	Acquire data assets & establishing context	Improve data quality & identify bias	Develop an appropriate learning system	Identify & Reduce error	Present results	Monitor change
PRINCIPLE	Purposeful	Openness	Multi-dimensional	Patterns & Trends	Counter-intuitive	Emergence	Adaptability
TOOLS	Archetypes Ladder of Inference	Iceberg Model	Stocks and Flows	Modeling & Simulation	Feedback Loops	Highest Leverage	Behavior Over Time
METRICS	Questions That Data Can Answer	Data Boundaries	Transparent open datasets	Experiments & Algorithms	Model Scores & Results	Predictions	Performance
INSIGHTS	Stakeholders	Data Owners	Data Managers	Engineers & Scientists	Engineers & Stakeholders	IT	Stakeholders
ARCHITECT-URE	USE CASES	DATA LAKE	DATA WAREHOUSE	SAFE LEARNING SPACE	QA/QC	PREDICTION ENGINE	DATA AUDITS

bT

SYSTEMS THINKING : PERSPECTIVES

THINKER	DEFINITION
Ross D. Arnold*, Jon P. Wade	Systems thinking is a set of synergistic analytic skills used to improve the capability of identifying and understanding systems, predicting their behaviors, and devising modifications to them in order to produce desired effects. These skills work together as a system Source: 2015, 2015 Conference on Systems Engineering Research. "A Definition of Systems Thinking: A Systems Approach" Ross D. Arnold*, Jon P. Wade
Barry Richmond	Barry Richmond, the originator of the systems thinking term, defines systems thinking as the art and science of making reliable inferences about behavior by developing an increasingly deep understanding of underlying structure10 (Richmond, 1994). He emphasizes that people embracing Systems Thinking position themselves such that they can see both the forest and the trees; one eye on each (Richmond, 1994
Peter Senge	Peter Senge defines systems thinking as a discipline for seeing wholes and a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static snapshots (Senge, 1990)
Linda Sweeney and John Sterman	Systems thinking is a management discipline that concerns an understanding of a system by examining the linkages and interactions between the components that comprise the entirety of that defined system.
Birgit Kopainsky, Stephen M. Alessi, and I. Davidsen	"Definition of systems thinking should include appreciation for long term planning, feedback loops, non-linear relationships between variables, and collaborative planning across areas of an organization"
Squires, Wade, Dominick, and Gelosha's definition	(per 2011 research & training project) Systems thinking is the ability to think abstractly in order to: incorporate multiple perspectives; work within a space where the boundary or scope of problem or system may be; understand diverse operational contexts of the system; identify inter- and intrarelationships and dependencies; understand complex system behavior; and most important of all, reliably predict the impact of change to the system

EXERCISE : ASK A WELL-FORMED QUESTION

WELL-FORMED QUESTION:

EXPERIENCE THE DATA	
TASK THE DECISION	
PERFORMANCE THE RESULT	
PERSPECTIVE THE DATA	

EXERCISE - IDENTIFY A MACHINE LEARNING PROJECT

ML?	PROJECT DESCRIPTION	INSIGHTS	COMPLEXITY	ACCURACY	SCALE	DATA ASSETS	RESOURCES	BOTTOM-LINE IMPACT